Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits.

1. Importing necessary packages and setting alias
   - import numpy as np
   - import pandas as pd
   - import matplotlib.pyplot as plt
2. Creating a numpy array
   - my_data = np.array([['Jan', 'Feb', 'Mar', 'Apr', 'May', 'June', 'July', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], [12, 13, 9, 8, 7, 8, 8, 7, 6, 5, 8, 10], [14, 16, 11, 7, 6, 6, 7, 6, 5, 8, 9, 12]])
3. Checking the shape of the array
   - np.shape(my_data)

```
1  np.shape(my_data)
```

```
(3, 12)
```

Dataframe

Pandas DataFrame is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the **data**, **rows**, and **columns**.

1. Now lets convert this numpy array into a dataframe
   - df = pd.DataFrame(my_data)

```
1 df
```

	0	1	2	3	4	5	6	7	8	9	10	11
0	Jan	Feb	Mar	Apr	May	June	July	Aug	Sep	Oct	Nov	Dec
1	12	13	9	8	7	8	8	7	6	5	8	10
2	14	16	11	7	6	6	7	6	5	8	9	12

   -

2. Transposing the data frame (Transposing is the process of changing the shape of dataframe by exchanging their place)
   - dataframe = df.transpose()

	0	1	2
0	Jan	12	14
1	Feb	13	16
2	Mar	9	11
3	Apr	8	7
4	May	7	6
5	June	8	6
6	July	8	7
7	Aug	7	6
8	Sep	6	5
9	Oct	5	8
10	Nov	8	9
11	Dec	10	12

3. Creating a column name for each column
   - dataframe.columns = ['Year', 'elec_01', 'elec_02']

```
1 dataframe
```

	Year	elec_01	elec_02
0	Jan	12	14
1	Feb	13	16
2	Mar	9	11
3	Apr	8	7
4	May	7	6
5	June	8	6
6	July	8	7
7	Aug	7	6
8	Sep	6	5
9	Oct	5	8
10	Nov	8	9
11	Dec	10	12

4. After transposing we will see that dataframes shape has been changed
   - dataframe.shape

```
1 dataframe.shape
```

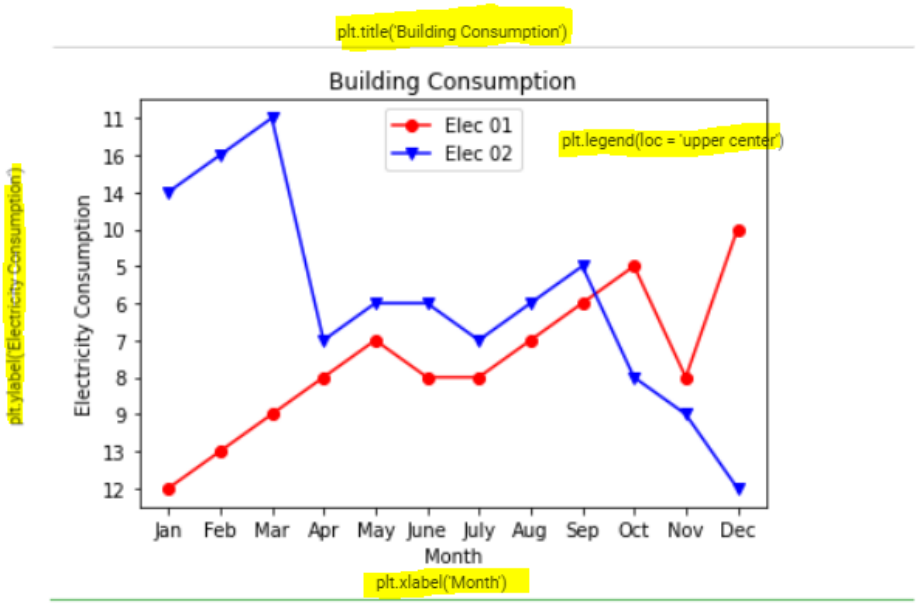
```
(12, 3)
```

Plotting datas

1. Plotting multiple sets of data to show relationship between datas

Description: here we are plotting two lines to show data's relationship.

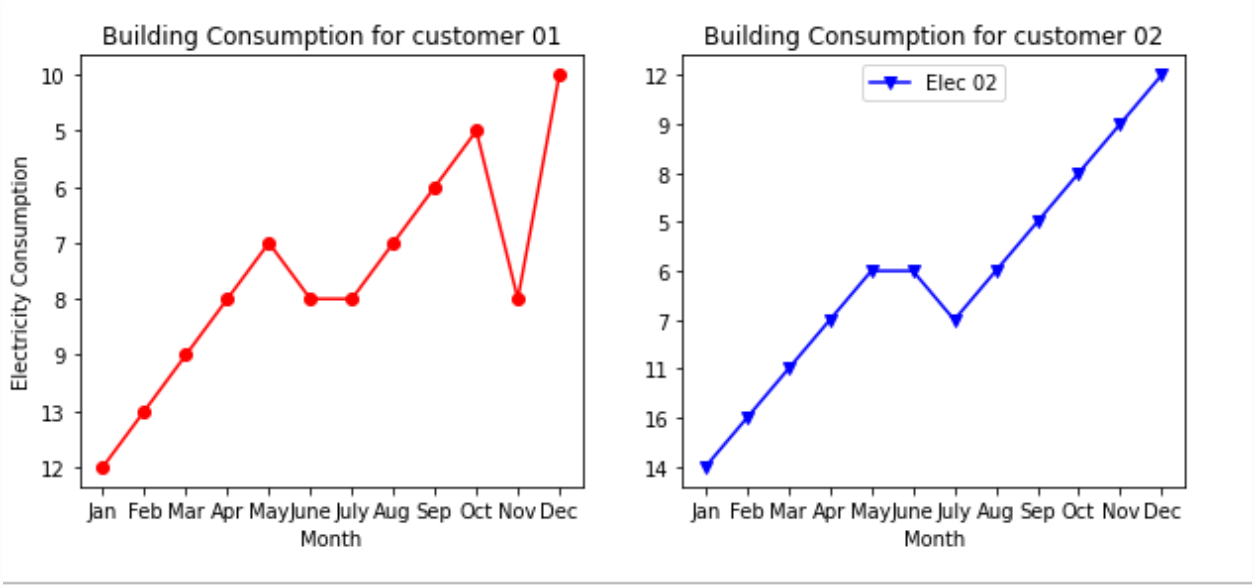
- plt.plot(dataframe.Year, dataframe.elec_01, color = 'red', label = 'Elec 01', marker = 'o')
- plt.plot(dataframe.Year, dataframe.elec_02, color = 'blue', label = 'Elec 02', marker = 'v')
- plt.xlabel('Month')
- plt.ylabel('Electricity Consumption')
- plt.title('Building Consumption')
- plt.legend(loc = 'upper center')
- plt.show()



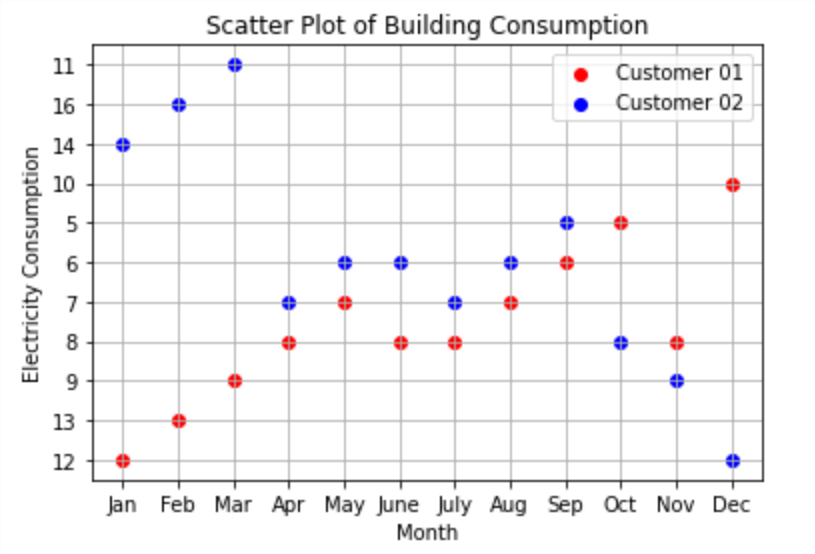
2. Plotting subplots to see data differently.
   Description: figure size is used to determine the size of the subplot area.
   - plt.figure(figsize=(10,4))
   - plt.subplot(1,2,1)
   - plt.plot(dataframe.Year, dataframe.elec_01, color = 'red', label = 'Elec 01', marker = 'o')
   - plt.xlabel('Month')
   - plt.ylabel('Electricity Consumption')
   - plt.title('Building Consumption for customer 01')

   - plt.subplot(1,2,2)
   - plt.plot(dataframe.Year, dataframe.elec_02, color = 'blue', label = 'Elec 02', marker = 'v')
   - plt.xlabel('Month')

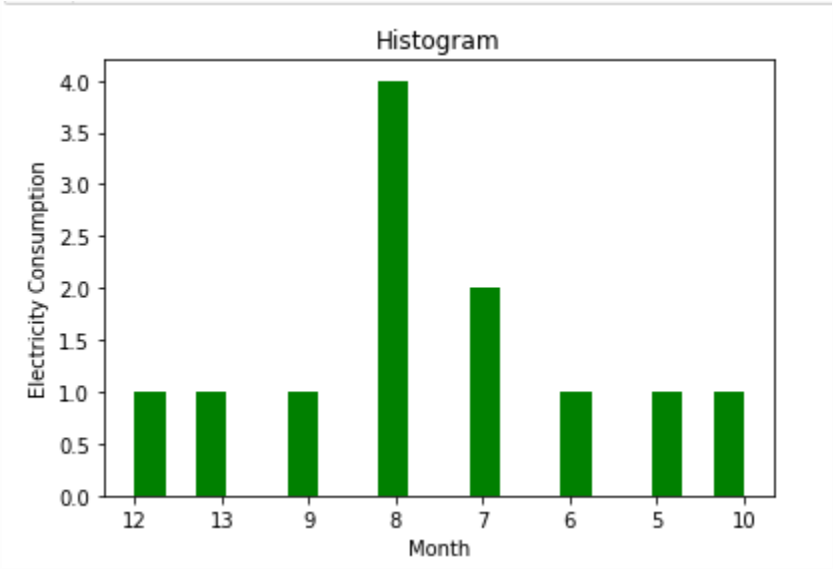
- plt.title('Building Consumption for customer 02')
- plt.show()



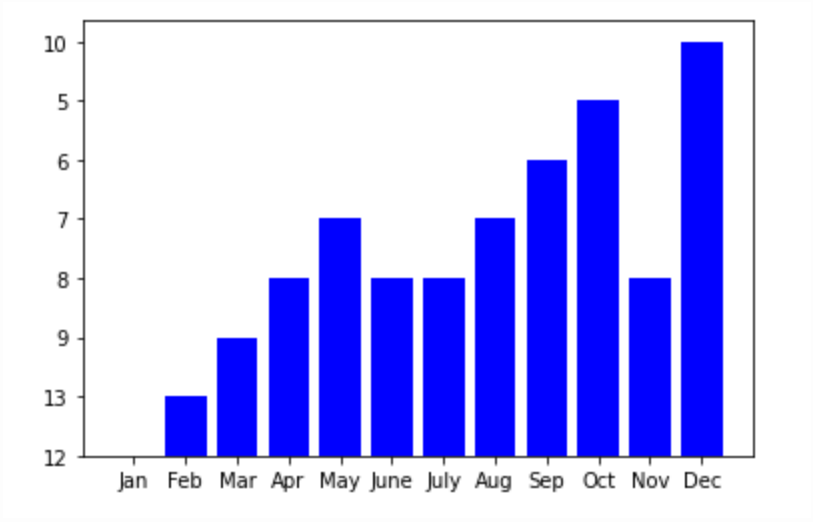
3. Plotting a scatter plots
   - plt.scatter(dataframe.Year, dataframe.elec_01, color = 'red', label = 'Customer 01')
   - plt.scatter(dataframe.Year, dataframe.elec_02, color = 'blue', label = 'Customer 02')
   - plt.xlabel('Month')
   - plt.ylabel('Electricity Consumption')
   - plt.title('Scatter Plot of Building Consumption')
   - plt.grid()
   - plt.legend(loc='best')
   - plt.show()



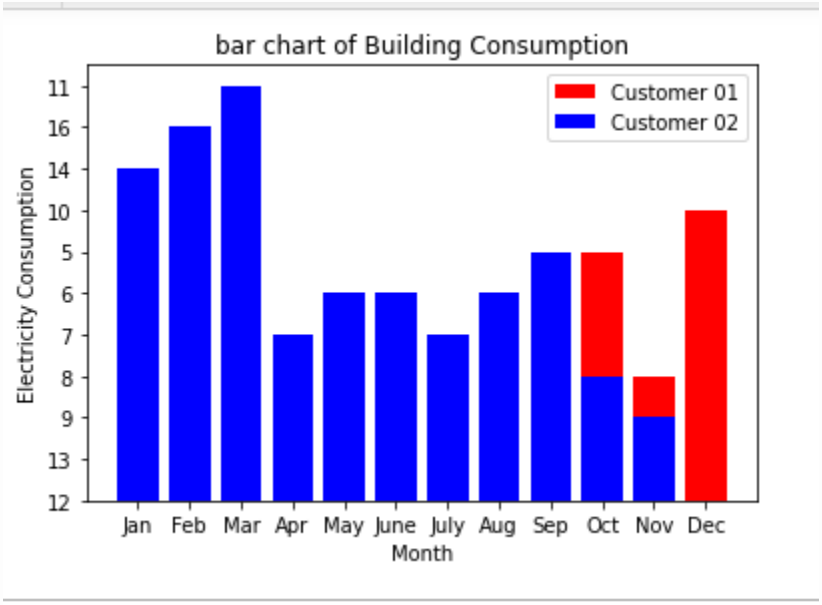
4. Plotting a histogram
   - plt.hist(dataframe.elec_01, bins=20, color = 'green')
   - plt.xlabel('Month')
   - plt.ylabel('Electricity Consumption')
   - plt.title('Histogram')
   - plt.show()



5. Plot a bar diagram
   - plt.bar(dataframe.Year, dataframe.elec_01, color = 'blue', width = 0.8)
   - plt.show()

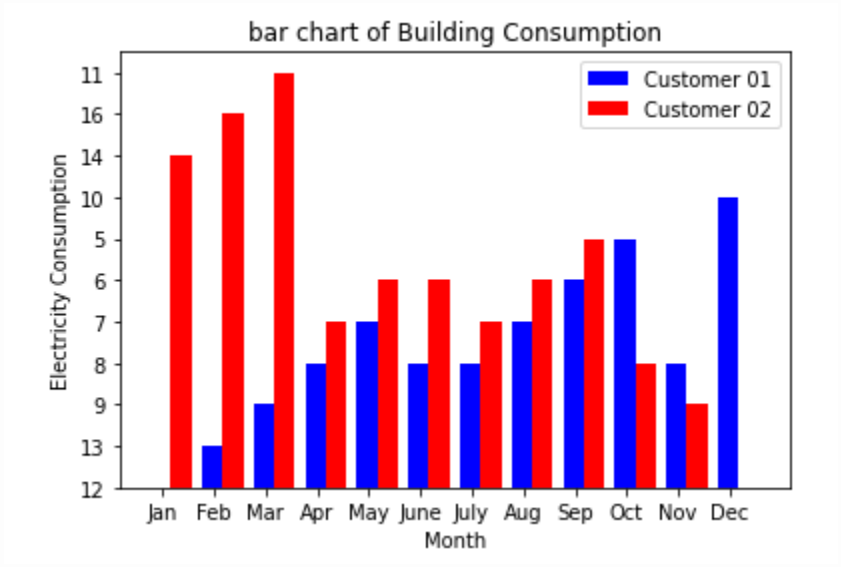


6. Double bar graph
    - plt.bar(dataframe.Year, dataframe.elec_01, color = 'red', label = 'Customer 01')
    - plt.bar(dataframe.Year, dataframe.elec_02, color = 'blue', label = 'Customer 02')
    - plt.xlabel('Month')
    - plt.ylabel('Electricity Consumption')
    - plt.title('bar chart of Building Consumption')
    - plt.legend(loc='best')
    - plt.show()



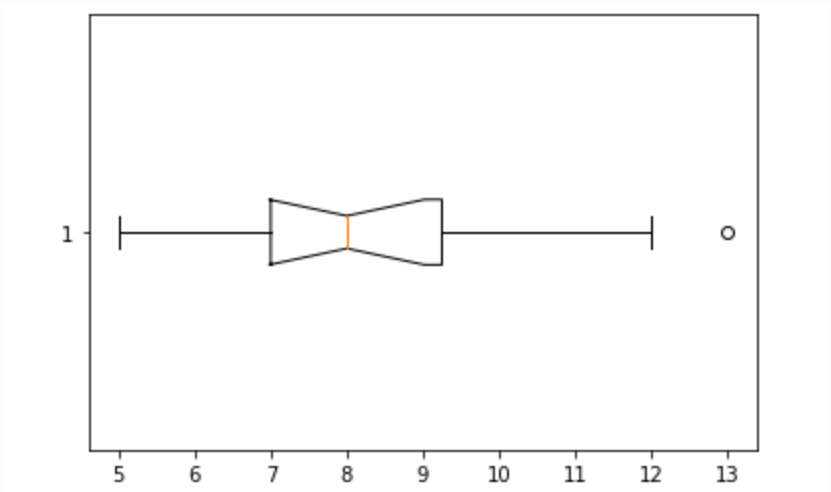
7. Bar Diagram with separate analysis

- bar_width = 0.4
- months_b = np.arange(12)
- plt.bar(months_b, dataframe.elec_01,bar_width, color = 'blue', label = 'Customer 01')
- plt.bar(months_b + bar_width, dataframe.elec_02, bar_width, color = 'red', label = 'Customer 02')
- plt.xticks(months_b + (bar_width)/12, ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'June', 'July', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'))
- plt.xlabel('Month')
- plt.ylabel('Electricity Consumption')
- plt.title('bar chart of Building Consumption'
- plt.legend(loc='best')
- plt.show()



8. Creating data for boxplot
   - customer_01 = [12, 13, 9, 8, 7, 8, 8, 7, 6, 5, 8, 10]

9. Plotting data in boxplot
   - plt.boxplot(customer_01, notch=True, vert= False)
   - plt.show()


1
5
6
7
8
9
10
11
12
13